Theoretical Computer Science 66 (1989) 333-340
North-Holland

NOTE

PROBABILISTIC TERMINATION VERSUS FAIR TERMINATION

Michael TIOMKIN*
Department of Computer Science, Technion-Israel Institute of Technology, Haifa 32000, Israel

Communicated by A. Meyer
Received May 1987
Revised August 1988

Abstract. In this note we show that probabilistic termination of concurrent programs is in many cases much simpler than the "fair" one. For a wide class of definitions of probabilistic termination we may express termination by Π_2^0 arithmetic formula, whereas the "fair" termination can be expressed only by Π_1^1 second-order arithmetic formula. Proof of "fair" termination usually needs induction on recursive ordinals, but proof of probabilistic termination has the complexity equivalent to that of deterministic program termination.

1. Introduction

The notion of *fairness* (cf. [2]) naturally arises while dealing with verification of concurrent programs. It is generally accepted that for proving concurrent program we may assume that the schedule has finite service time, i.e. if a process is enabled long enough, it should be chosen. There exist many different notions of "fairness", e.g. the usual, the extreme and the absolute fairness, etc. As was pointed out by Hart, Sharir and Pnueli, all of these notions are based on different classes of possible deterministic schedulers (cf. [6]), or, equivalently, on different sets of infinite paths in the tree of all possible choices made by the schedulers. For every notion of "fairness", a program is said to be "fair" terminating, if the set of its infinite computations does not contain "fair" paths.

One of the objections to the theories of "fair" computations is their high complexity. Thus, "fair" termination of a recursive concurrent program is Π_1^1 complete, and proving such termination involves induction on countable recursive ordinals (cf. [3, 5], and [10, Chapter 16]), whereas termination of a recursive deterministic program may be expressed by a Π_2^0 arithmetic formula (see the remark below).

Alternatively, many papers appeared in the last years consider probabilistic schedules (cf. [6, 9, 13]). That is to say, instead of the class of admissible determinstic

* Current address: IBM Science and Technology, Technion City, Haifa 32000, Israel.

0304-3975/89/$3.50

schedules for the "fairness", one can easily define the class of admissible probabilistic schedules, and instead of one fixed set of infinite "fair" paths one can deal with sets of paths which have probability 1 for every "admissible" probabilistic schedule. We shall say that a program terminates *almost everywhere* (a.e.), if the set of its finite computations has a measure 1 for every "admissible" probabilistic schedule. Then one can easily prove that "fair" termination implies a.e. termination for different sets of schedules. For example, if the only admissible schedule chooses the possible processes with equal probabilities, then "equifair" termination (and almost all other fair terminations) implies a.e. termination (cf. [4]). If we allow schedules choosing any possible process with probability greater than ε for some $\varepsilon > 0$, then "fair" ("extremely fair") termination implies a.e. termination (cf. [9, 13]).

The reason for dealing with probabilistic schedulers is as follows. First, in the real system the "closed world" assumption is not exactly true. Any real scheduler (operating system) depends on the events from the external world, such as the speed of channels and processors, the input data from measuring devices, and, the worst, human interface. This makes the probabilistic analysis of system behavior so attractive. Second, after years of extensive (and successful!) using pseudo-random numbers, one might feel that a compound digital-analogue machine with real random number generation might be much more convenient. Clearly, we cannot promise the exact probability distribution we need (recall that here we deal with an analogue device). We may be assured only that the distribution is "close" enough. This brings us to the notion of "stable" probability distribution (cf. [13]), where the probability of choosing a process must be "far enough" from 0.

However, many computer scientists still prefer discrete versions of program semantics and verification. The purpose of this note is to show that, in the case of verification of concurrent programs, the non-discrete reasoning is much simpler than the discrete one. We shall see that for "simply definable" classes of probabilities the corresponding a.e. termination is Π_2^0. Thus a.e. termination of a concurrent program and termination of a deterministic program are of the same complexity.

In detail, this note is organized as follows. In Section 2 we give some basic definitions and notation. Section 3 contains the main results and a short discussion dealing with sets of probabilities having higher degree of definability.

2. Definitions and notation

We introduce here the habitual definitions of a computation tree for a finite number of concurrent sequential processes, or, equivalently, finitely nondeterministic program. In addition, we define some (topo)logical and probabilistic notions which will be used in the sequel.

For a finitely nondeterministic program with fixed input data ξ (cf. the definition of the Guarded Commands language *GC* in [1] or [2]), we use the standard definition of its computation tree $T(\xi)$ (cf. [3] or [2, p. 10]). We can easily extend this Guarded

Commands language with the usual deterministic iteration constructs, such as **while** or **repeat**. Also, we can delete the "deterministic" nodes of out-degree 1 making $T(\xi)$ a recursively enumerable (RE) tree instead of the primitive recursive one. This is similar to what is done in biology, economics and sociology, treating differently the stable and the unstable stages of the process (cf. [8]). Then a computation tree will have three kinds of nodes (states)—the terminating nodes, the transient ones, and the nodes where a program loops deterministically. All the results remain unchanged in this case. This computation tree has a finite maximal out-degree (bounded by the maximal number of guards in the selection statements). Moreover, for a node σ with out-degree greater than 0, the exact out-degree of σ can be recursively computed. Obviously, the tree $T(\xi)$ with the out-degree $\leq n$ can be recursively mapped to an RE subtree of a full n-tree, $T_n = \{1, \ldots, n\}^{<\omega}$.

A computation of a program ξ is a maximal path in the tree $T(\xi)$. There exists a natural topology on the set $T^{\max}(\xi)$ of all maximal paths on $T(\xi)$. This topology is generated by the clopen (i.e. closed and open) sets $U_\sigma = \{p \in T^{\max}(\xi): p$ passes through $\sigma\}$, where σ is a node of $T(\xi)$. For a program ξ, the set of infinite computations of ξ, i.e., the set of maximal infinite paths in the corresponding $T(\xi)$, is closed in this topology. The set of ξ's terminating states $Term_\xi$ is an RE set of nodes of $T(\xi)$. Further on we shall deal with the open set $TERM_\xi = \bigcup_{\sigma \in Term_\xi} U_\sigma \subseteq T^{\max}(\xi)$ generated by the set of terminating states of ξ. All these U_σ are mutually disjoint one-element clopen sets. Notice that the set $TERM_\xi$ is not always complementary to the set of infinite paths of ξ: there may be states where ξ loops deterministically.

We also give some definitions concerning probabilities on the Borel subsets of $T^{\max}(\xi)$.

We define probability as a nonnegative measure which is equal to 1 on the whole space. $Pr(T^{\max}(\xi))$ is the set of probabilities on the Borel subsets of $T^{\max}(\xi)$. Every measure on Borel subsets of a topological space is uniquely determined by its values on the base open sets. In our case, every probability from $Pr(T^{\max}(\xi))$ is uniquely determined by its values on the sets U_σ. These values, in their turn, are uniquely determined by the conditional probabilities to pass from a node of $T(\xi)$ to its immediate successor. Actually, the above conditional probabilities are defined on $E(T(\xi))$, the set of edges of $T(\xi)$. Sometimes a conditional probability is not defined (the probability to arrive to a node is 0). In this case we can assume it to be equal to any value from the segment [0, 1], provided that the sum of the probabilities to exit a node is equal to 1.

So every probability from $Pr(T^{\max}(\xi))$ may be naturally mapped ino the space $Ms_\xi = [0, 1]^{E(T(\xi))}$.

The set Ms_ξ with the product topology (generated by the finite products) is a compact (as a product of compacts, cf. [11, Section 9.4.19, p. 166]). The set of probabilities is defined by the demand that for every node σ the sum of conditional probabilities on edges exiting σ is equal to 1. Thus the set of probabilities is a closed subset of Ms_ξ, and hence it is a compact as well (cf. [11, Section 9.1.2, p. 158]). In the next section we only use the compactness of $Pr(T^{\max}(\xi))$ and the trivial fact that the value of a probability on U_σ is a continuous function on Ms_ξ (it depends continuously on conditional probabilities on a fixed finite number of edges from $E(T(\xi))$.

The (uniform) a.e. termination is defined as usual.

Definition 2.1. For a set of probabilities $SP \subseteq Pr(T^{\max}(\xi))$, we shall say that a program ξ *terminates SP almost everywhere* (*a.e.*) if, for every probability $p \in SP$, $p(TERM_\xi) = 1$.

Usually, this set of probabilities SP depends on a program ξ. One can find examples of SP a.e. termination in [13] (the "pseudoprobabilistic" termination), or [9] (the "P-valid" termination). Since we are interested here in the definability properties of different notions of termination, we shall deal only with sets of probabilities which are definable in some theory.

Now it is time for some logic. We introduce definable sets of probabilities from $Pr(T^{\max}(\xi))$. Recall that the letter "F" usually denotes a closed set, "G" an open one, F_σ a countable union of closed sets, etc. (cf. [11]).

Definition 2.2. We shall say that the set $SP \subseteq Ms_\xi$ *has a closed recursive algebraic definition Φ*, or shortly *is an RAF set*, if

(1) SP is closed.

(2) Φ is a recursive function defined on finite sets of edges of $E(T(\xi))$ (notice that such sets are constructive objects). For every finite set of edges $Ed \subseteq E(T(\xi))$, $\Phi(Ed)$ is a formula $\varphi(x_1, \ldots, x_{|Ed|})$ in real analysis (theory of real numbers with operators "+" and "*", predicates "=" and "<", and variables and quantifiers over reals, cf. [12]).

(3) For every finite set of edges $Ed \subseteq E(T(\xi))$ the projection of SP on Ed, i.e. the set $\{f \in [0, 1]^{Ed} : f$ may be extended to some $p \in SP\}$ is definable by $\Phi(Ed)$. Notice that by compactness of Ms_ξ and (1) this projection is a closed subset of $[0, 1]^{Ed}$ (as a continuous image of compact, cf. [11, Section 9.1.4, p. 158]).

Remark that a closed set SP may be easily reconstructed from its projections; it is equal to the intersection of the cylinders raised from its projections.

The following proposition shows that the set of probabilities is a well-defined subset of Ms_ξ.

Proposition 2.3. *The set $Pr(T^{\max}(\xi))$ is an RAF set.*

Proof. For a finite set of edges $Ed \subseteq E(T(\xi))$ we construct a formula defining the projection of $Pr(T^{\max}(\xi))$ on Ed. We define the set of the left nodes of Ed as $Le(Ed) = \{\sigma \in T(\xi) : \langle \sigma, \sigma' \rangle \in Ed$ for some $\sigma'\}$. For a node $\sigma \in Le(Ed)$ the out-degree d_σ in $T(\xi)$ is greater than 0, hence this degree may be obtained recursively from ξ

and σ. For a node $\sigma \in Le(Ed)$ we define also the nonempty set of edges $Ed_\sigma = \{e \in Ed: e = \langle\sigma, \sigma'\rangle$ for some $\sigma'\} \subseteq Ed$. We can define the projection of $Pr(T^{\max}(\xi))$ on the set E_σ by the formula

$$\psi_\sigma = \begin{cases} \sum_{e \in Ed_\sigma} p(e) \leq 1 & \text{if } |Ed_\sigma| < d_\sigma, \\ \sum_{e \in Ed_\sigma} p(e) = 1 & \text{if } |Ed_\sigma| = d_\sigma. \end{cases}$$

Then the projection of $Pr(T^{\max}(\xi))$ on the set Ed is defined by the formula $\&_{\sigma \in Le(Ed)} \psi_\sigma$. This formula depends recursively on program ξ and set Ed. □

Because intersection of two RAF sets is an RAF set, Proposition 2.3 implies the following corollary.

Corollary 2.4. *If a set $SP \subseteq Ms_\xi$ is an RAF set, then the set of probabilities from SP, i.e. the set $SP \cap Pr(T^{\max}(\xi))$, is also an RAF set.*

Recall that because of the finite encoding of RAF sets, one can easily define countable unions, intersections, etc., of uniformly defined families of RAF sets.

Definition 2.5. We shall say that a set $SP \subseteq Ms_\xi$ *has a recursive algebraic F_σ definition* Φ, or shortly *is an RAF$_\sigma$ set*, if Φ is a recursive function defined on natural numbers, and $SP = \bigcup_n SP_n$, where $\Phi(n)$ is a closed recursive algebraic definition of SP_n.

For instance, one can see that the set of "stable" probabilities (cf. [13]), i.e. probabilities with $p(e) > \varepsilon > 0$ for every $e \in E(T(\xi))$, this set is an RAF$_\sigma$ set. The set consisting of one probability with recursively computable algebraic values is trivially an RAF set.

In the next section we shall deal with expressibility of a.e. termination for RAF$_\sigma$ sets of probabilities.

3. The main results

We are going to prove that a.e. termination for an RAF$_\sigma$ set of probabilities may be expressed by a Π_2^0 formula. For this we shall use the following well-known fact from mathematical analysis.

Fact 3.1. *If $\{f_n\}$ is a sequence of continuous real functions on a compact K, and $\{f_n\}$ converges pointwise monotonically to a continuous function f on K, then the sequence $\{f_n\}$ converges to f uniformly on K.*

For the proof, note that then the sequence $|f-f_n|$ decreases pointwise monotonically to 0, and we can apply Dini's Theorem, cf. [11, Section 9.2.11, p. 162].

From this fact we obtain the following corollary.

Corollary 3.2. *Let $SP \subseteq Pr(T^{\max}(\xi))$ be an RAF set of probabilities. Then the SP a.e. terminating of a program ξ with fixed input data can be expressed by a Π_2^0 formula. This formula depends recursively on the definition of SP and on ξ.*

Proof. The set of terminating states of ξ is RE. Let $term_k$ be the finite set of terminating states computed by the enumerating process after k steps of computation. We have, trivially, that $term_k \subseteq term_{k+1} \subseteq Term_\xi$, and $Term_\xi = \bigcup_k term_k$. The value of $term_k$ depends recursively on k and ξ. Now for an integer k, we define a function f_k on $Pr(T^{\max}(\xi))$ by

$$f_k(p) = p\Bigl(\bigcup_{\sigma \in term_k} U_\sigma\Bigr) = \sum_{\sigma \in term_k} p(U_\sigma),$$

for a probability $p \in Pr(T^{\max}(\xi))$. The sequence of functions $\{f_k\}$ converges pointwise monotonically to $p(TERM_\xi)$. Also, the RAF set SP is a compact as a closed subset of $Pr(T^{\max}(\xi))$. Hence if ξ is SP a.e. terminating, i.e. the functions $\{f_k\}$ converge pointwise to 1 on SP, then, by Fact 3.1, they converge to 1 uniformly. Thus we can express the SP a.e. terminating of ξ by the expression

$$\forall m\, \exists k\, \forall p \in SP \quad p\Bigl(\bigcup_{\sigma \in term_k} U_\sigma\Bigr) > 1 - \frac{1}{m}.$$

Notice that the probability $p(\bigcup_{\sigma \in term_k} U_\sigma)$ depends only on a finite number of values of p on $E(T(\xi))$. Projections of SP on a finite number of edges on $E(T(\xi))$ are expressible in real analysis. Thus, because the validity of a formula from real analysis is recursive (cf. [12]), we can recursively test the validity of the formula $\forall p \in SP\, p(\bigcup_{\sigma \in term_k} U_\sigma) > 1 - 1/m$ for every given pair m, k. □

Now, using Corollary 3.2 one can easily prove the same result for RAF_σ sets of probabilities and programs with variable input data.

Theorem 3.3. *For an RAF_σ set of probabilities $SP \subseteq Pr(T^{\max}(\xi))$, and a program ξ with RE set of finite possible input data, the SP a.e. terminating of a program ξ for every input is expressible by a Π_2^0 formula.*

Proof. Let SP be an RAF_σ subset of $Pr(T^{\max}(\xi))$. By Definition 2.5, SP is equal to $\bigcup_k SP_k$, where SP_k is a uniformly defined RAF subset of $Pr(T^{\max}(\xi))$. Then the following two conditions are equivalent:

(1) for every $p \in SP$, $p(TERM_\xi) = 1$;

(2) for every $p \in SP_k$, $p(TERM_\xi) = 1$.

Therefore we can express the SP a.e. termination of ξ by the following statement:

"for every possible input Inp, and for every integer k, $\xi(Inp)$ is SP_k a.e. terminating".

By Corollary 3.2, this expression is Π_2^0 (it is a Π_2^0 expression with two additional universal quantifiers from the outside). □

Remark. Note that the usual termination of a *deterministic* program ξ for every input has the same complexity (Π_2^0), because it is expressed as follows:

"for every possible input *Inp*, there exists a k such that $\xi(Inp)$ terminates after k steps".

In many cases this theorem allows us to prove that probabilistic termination is Π_2^0. For instance, we say that a probability p is *stable* if there exists an $\varepsilon > 0$, such that $p(e) > \varepsilon$ for all $e \in E(T(\xi))$ (cf. [13]). We define that a program ξ terminates "pseudoprobabilistically" (cf. [13]) if the set $TERM_\xi$ has probability 1 for all stable probabilities. Then we have the following corollary.

Corollary 3.4. *The "pseudoprobabilistic" termination is* Π_2^0.

Proof. The "pseudoprobabilistic" termination is the a.e. termination for the set of stable probabilities. The set of "stable" probabilities is RAF_σ, and the result follows from Theorem 3.3. □

As it was pointed out in Section 2, one can easily define recursive algebraic Borel subsets of $Pr(T^{\max}(\xi))$ (or Ms_ξ) of any finite (and even infinite recursive) degree. We can define an "open" recursive algebraic set (or shortly an *RAG* set) as a complement to an RAF set, and similarly, $\mathrm{RAF}_{\sigma\delta}$ and RAG_δ sets etc. One can easily see that every RAG set is also an RAF_σ set, hence Theorem 3.3 holds for RAG sets as well.

Unfortunately, the method used in the proof of Theorem 3.3 is not applicable for the sets of the degree higher than RAF_σ. In this connection, we would conjecture the following.

Conjecture 3.5. *For a program ξ, we can construct a $RAF_{\sigma\delta}$ (RAG_δ) subset $SP(\xi)$ of $Pr(T^{\max}(\xi))$ such that the $SP(\xi)$ a.e. termination of a program ξ is Π_1^1 complete.*

Acknowledgment

The author is very grateful to M. Kaminski (Technion, Haifa) for fruitful discussions and critical reading the paper. Special thanks are due to an anonymous referee for useful suggestions which considerably improved the readability of the paper.

References

[1] E.W. Dijkstra, Guarded commands, nondeterminacy and formal derivation of programs, *Comm. ACM* 18 (1975) 453-457.

[2] N. Francez, *Fairness* (Springer, New York, 1986).

[3] O. Grumberg, N. Francez, J.A. Makowsky and W.P. de Roever, A proof rule for fair termination of guarded commands, *Inform. Contr.* 66 (1985) 83-102.

[4] O. Grumberg, Fair termination of programs, Ph.D. Thesis, Computer Science Department, Technion, Haifa 1984 (in Hebrew).

[5] D. Harel, Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness. *J. ACM* 33 (1986) 224-248.

[6] S. Hart, M. Sharir and A. Pnueli, Termination of probabilistic concurrent programs, *ACM Trans. on Programming Languages and Systems* 5 (1983) 356-380.

[7] J. Neveu, *Mathematical Foundations of the Calculus of Probability* (Holden-Day, San Francisco, CA, 1965).

[8] G. Nikolis and I. Prigogin, *Self-organization in Nonequilibrium Systems* (Wiley-Interscience, New York, 1977).

[9] A. Pnueli, On the extremely fair treatment of probabilistic algorithms, in: *Proc. 15th Symp. Theory of Computing*, Boston (1983) 278-290.

[10] H. Rogers, *Theory of Recursive Functions and Effective Computability* (McGraw-Hill, New York, 1967).

[11] H.L. Royden, *Real Analysis* (McMillan, New York, 1968, 2nd ed.).

[12] A. Tarsky, *A Decision Method for Elementary Algebra and Geometry*, (Berkeley and Los Angeles, 1951, 2nd ed.).

[13] M.L. Tiomkin and J. A. Makowsky, Probabilistic propositional dynamic logic, Tech. Report TR#305, Dept. of Computer Science, Technion, Haifa, 1983.